

19 RÉPUBLIQUE FRANÇAISE  
INSTITUT NATIONAL  
DE LA PROPRIÉTÉ INDUSTRIELLE  
PARIS

11 N° de publication :  
(à n'utiliser que pour les  
commandes de reproduction)

2 822 627

21 N° d'enregistrement national : 01 03909

51 Int Cl<sup>7</sup> : H 04 Q 7/32

12 DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 22.03.01.

30 Priorité :

43 Date de mise à la disposition du public de la  
demande : 27.09.02 Bulletin 02/39.

56 Liste des documents cités dans le rapport de  
recherche préliminaire : *Se reporter à la fin du  
présent fascicule*

60 Références à d'autres documents nationaux  
apparentés :

71 Demandeur(s) : WAVECOM SA Société anonyme —  
FR.

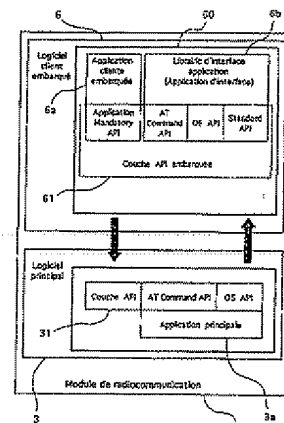
72 Inventeur(s) : MONTES JACQUES.

73 Titulaire(s) :

74 Mandataire(s) : CABINET PATRICE VIDON.

54 MODULE DE RADIOCOMMUNICATION HEBERGEANT ET EXECUTANT UN LOGICIEL CLIENT, ET PROCEDE CORRESPONDANT DE MISE EN OEUVRE D'UN LOGICIEL CLIENT DE PILOTAGE.

57 L'invention concerne un module de radiocommunication, du type hébergeant et exécutant un logiciel principal assurant notamment des fonctions de radiocommunication; Le logiciel principal comprend des moyens d'exécution de commandes de pilotage, envoyées au logiciel principal par au moins un logiciel client de pilotage et appartenant à un jeu de commandes de pilotage prédéterminé. Selon l'invention, le module de radiocommunication héberge et exécute en outre au moins un logiciel client, dit logiciel client embarqué. Le logiciel client embarqué et le logiciel principal comprennent des moyens permettant au logiciel client embarqué de jouer au moins un des deux rôles suivants: - le rôle d'un logiciel client de pilotage, envoyant des commandes de pilotage au logiciel principal, et recevant du logiciel principal des réponses, résultant de l'exécution de certaines des commandes de pilotage; - le rôle d'un logiciel client de supervision, gérant l'exécution de commandes de pilotage envoyées par un logiciel client de pilotage, dit logiciel client externe, hébergé et exécuté par un équipement tiers coopérant avec le module de radiocommunication.



FR 2 822 627 - A1



**Module de radiocommunication hébergeant et exécutant un logiciel client, et procédé correspondant de mise en œuvre d'un logiciel client de pilotage.**

Le domaine de l'invention est celui des systèmes de radiocommunication notamment, mais non exclusivement, selon le standard GSM (pour "Global System for Mobile communications" en anglais), DCS 1800 (pour "Digital Cellular System 1800 MHz" en anglais), PCS 1900 (pour "Personal Communication System" en anglais), GPRS (pour "General Packet Radio service"), ou encore UMTS (pour "Universal Mobile Telecommunication System" en anglais).

Plus précisément, l'invention concerne un module de radiocommunication. On rappelle que le module de radiocommunication est un élément essentiel d'un radiotéléphone. Il héberge et exécute un logiciel principal (communément appelé "logiciel de radiocommunication" ou "logiciel GSM") qui assure notamment des fonctions de communication sans fil (radiocommunication) et pilote divers éléments matériels (écran, clavier, haut-parleur, ...) du radiotéléphone.

Habituellement (première application), le module de radiocommunication est compris dans un terminal (ou ME, pour "Mobile Equipment" en anglais) qui coopère avec une carte SIM (pour "Subscriber Identity Module" en anglais, ou module d'identification d'abonné).

D'autres applications ont d'ores et déjà été envisagées pour le module de radiocommunication précité.

Il a notamment été proposé (seconde application) d'intégrer le module de radiocommunication dans des dispositifs autres que des terminaux de radiocommunication, mais nécessitant néanmoins une fonctionnalité de communication sans fil. A titre d'exemple, on peut citer des dispositifs de télémétrie (relevés de compteur), des dispositifs d'alarme ou encore des dispositifs lecteur de cartes bancaires.

Il a également été proposé (troisième application) de fournir le module de radiocommunication sous forme indépendante : il est alors qualifié de modem. Un tel modem ne comprend aucun élément matériel (écran, clavier, haut-parleur, ...) d'interface homme/machine. Il est destiné à coopérer avec un équipement tiers (supportant un logiciel client), qui lui possède des éléments matériels d'interface homme/machine. L'équipement tiers peut notamment, mais non exclusivement, être un micro-ordinateur.

D'une façon générale, l'équipement tiers héberge et exécute un logiciel client de pilotage, permettant de piloter (c'est-à-dire de commander) le module de radiocommunication, grâce à un jeu de commandes de pilotage au format AT. Les commandes AT (pour "ATtention command" en anglais) permettent à l'équipement tiers (ou TE pour "Terminal Equipment") d'exiger du terminal de radiocommunication auquel il est relié d'exécuter certaines actions prédéterminées. A cet effet, le logiciel principal (hébergé par le module de radiocommunication) comprend des moyens d'exécution des commandes AT, qui lui sont envoyées par le logiciel client de pilotage (hébergé par l'équipement tiers).

Pour plus de précisions concernant les commandes AT, on pourra se reporter d'une part aux normes "GSM 07.05" et "GSM 07.07" de l'ETSI et d'autre part à la recommandation V25ter de l'ITU-T, qui sont insérées ici par référence.

D'une façon générale, le pilotage du module de radiocommunication par un équipement tiers, grâce aux commandes AT, ne s'applique pas uniquement dans le cadre de la première application précitée (module de radiocommunication formant modem), mais aussi dans le cadre des première et seconde application précitées (module de radiocommunication compris dans un terminal de radiocommunication ou un autre dispositif).

En d'autres termes, quelle que soit l'application envisagée, le module de radiocommunication peut être piloté par un équipement tiers avec lequel il coopère (généralement via une liaison série). Dans ce cas, un logiciel client de pilotage (comprenant une "application cliente externe"), hébergé et exécuté par l'équipement tiers, envoie des commandes AT à un logiciel principal, hébergé et exécuté par le module de radiocommunication, en vue de leur exécution par ce dernier.

Comme illustré sur la figure 2, le fonctionnement de la technique actuelle de pilotage d'un module de radiocommunication par un équipement tiers peut être résumé ainsi :

- étape "1" : l'application cliente externe (logiciel client de pilotage) 2 envoie une commande AT ;

- étape "2" : la liaison série 5 transmet la commande AT aux moyens 4 d'exécution des commandes AT compris dans le logiciel principal 3 hébergé et exécuté par le module de radiocommunication 1 ;
- étape "3" : les moyens 4 d'exécution exécutent la commande AT ;
- 5 - étape "4" : après exécution, les moyens d'exécution 4 envoient une réponse AT à l'application cliente externe 2 ;
- étape "5" : cette réponse est envoyée à travers la liaison série 5 ;
- étape "6" : l'application cliente externe 2 reçoit la réponse.

10 Chacune de ces étapes est illustrée sur la figure 2 par un cercle dans lequel est inscrit le numéro de l'étape concernée. La même convention est adoptée dans les figures suivantes relatives à la présente invention (et qui sont décrites en détail dans la suite de la description).

La technique actuelle de pilotage d'un module de radiocommunication par un équipement tiers présente plusieurs inconvénients.

15 Tout d'abord, elle nécessite un double jeu de ressources (processeur et mémoire). En effet, le module de radiocommunication comprend un processeur et une mémoire (premier jeu de ressources) et l'équipement tiers dispose également d'un processeur et d'une mémoire (second jeu de ressources). La technique actuelle précitée est donc coûteuse en terme de matériel et de consommation énergétique.

20 Un autre inconvénient de la technique actuelle précitée est que le module de radiocommunication subit entièrement le pilotage effectué par l'équipement tiers. En effet, le logiciel client de pilotage, hébergé et exécuté par l'équipement tiers, est le "maître", tandis que le logiciel principal, hébergé et exécuté par le module de radiocommunication, est l'"esclave".

25 L'invention a notamment pour objectif de pallier ces inconvénients de l'état de la technique.

Plus précisément, l'un des objectifs de la présente invention est de fournir une technique de pilotage d'un module de radiocommunication, du type mettant en œuvre un logiciel client de pilotage, qui soit simple et peu coûteuse (en terme de matériel et de consommation énergétique).

30

L'invention a également pour objectif de fournir une telle technique de pilotage permettant au module de radiocommunication, lorsque le pilotage est effectué par un équipement tiers, de pouvoir superviser (y inclus agir sur) le déroulement de ce pilotage. En d'autres termes, on souhaite que le module de radiocommunication ne joue pas  
5 uniquement un rôle d'esclave.

Ces différents objectifs, ainsi que d'autres qui apparaîtront par la suite, sont atteints selon l'invention à l'aide d'un module de radiocommunication, du type hébergeant et exécutant un logiciel principal assurant notamment des fonctions de radiocommunication, ledit logiciel principal comprenant des moyens d'exécution de  
10 commandes de pilotage, envoyées au logiciel principal par au moins un logiciel client de pilotage et appartenant à un jeu de commandes de pilotage prédéterminé. Selon l'invention, le module de radiocommunication héberge et exécute en outre au moins un logiciel client, dit logiciel client embarqué. En outre, le logiciel client embarqué et le logiciel principal comprennent des moyens permettant au logiciel client embarqué de  
15 jouer au moins un des deux rôles suivants :

- le rôle d'un logiciel client de pilotage, envoyant des commandes de pilotage au logiciel principal, et recevant du logiciel principal des réponses, résultant de l'exécution de certaines des commandes de pilotage ;
- le rôle d'un logiciel client de supervision, gérant l'exécution de commandes de  
20 pilotage envoyées par un logiciel client de pilotage, dit logiciel client externe, hébergé et exécuté par un équipement tiers coopérant avec le module de radiocommunication.

Le principe général de l'invention consiste donc à héberger sur le module de radiocommunication au moins un logiciel client pouvant jouer le rôle d'un logiciel client  
25 de pilotage et/ou le rôle d'un logiciel client de supervision.

Ainsi, dans le cas où le logiciel client embarqué joue le rôle d'un logiciel client de pilotage, le module de radiocommunication présente un fonctionnement autonome et peu coûteux. En effet, dans ce cas, le module de radiocommunication n'a pas à coopérer  
avec un équipement tiers, et le logiciel principal et le logiciel client de pilotage utilisent  
30 les mêmes ressources (même processeur et même mémoire).

La présente invention permet également, dans le cas où le logiciel client embarqué joue le rôle d'un logiciel client de supervision, de ne pas limiter le module de radiocommunication à un rôle d'esclave vis-à-vis de l'équipement tiers qui exécute le logiciel client de pilotage. En effet, le logiciel client de supervision, exécuté par le module de radiocommunication, gère le pilotage demandé par le logiciel client de pilotage, exécuté par l'équipement tiers. On notera que, dans ce cas, le logiciel client embarqué est un logiciel supplémentaire par rapport à la configuration de l'état de la technique précitée. Toutefois, ce logiciel supplémentaire est peu coûteux puisqu'il utilise les mêmes ressources (processeur et mémoire) que le logiciel principal également hébergé par le module de radiocommunication.

Avantageusement, pour permettre au logiciel client embarqué de jouer le rôle d'un logiciel client de supervision :

- le logiciel principal comprend des moyens d'aiguillage de commandes, en fonction d'une politique d'aiguillage de commandes déterminée, de façon à transmettre les commandes de pilotage provenant du logiciel client externe vers le logiciel client embarqué et/ou vers les moyens d'exécution compris dans le logiciel principal ;

- le logiciel client embarqué comprend des moyens de traitement des commandes de pilotage aiguillées vers lui par lesdits moyens d'aiguillage de commandes.

De cette façon, le logiciel client embarqué peut :

- soit n'effectuer aucune action, les commandes étant transmises uniquement au logiciel principal, qui les exécute directement ;
- soit effectuer un filtrage des commandes qui lui sont transmises sans être exécutées par le logiciel principal. Le logiciel client embarqué peut ainsi par exemple décider, en fonction de critères déterminés, quelles commandes doivent être exécutées par le logiciel principal et quelles commandes peuvent faire l'objet d'une réponse sans exécution ;

- soit un "espionnage" des commandes dont il reçoit une copie et qui sont par ailleurs directement exécutées par le logiciel principal.

De façon avantageuse, pour permettre au logiciel client embarqué de jouer le rôle d'un logiciel client de supervision :

- le logiciel principal comprend des moyens d'aiguillage de réponses, en fonction d'une politique d'aiguillage de réponses déterminée, de façon à transmettre des réponses, résultant de l'exécution de certaines des commandes de pilotage par les moyens d'exécution compris dans le logiciel principal, vers le logiciel client embarqué et/ou vers le logiciel client externe;
- le logiciel client embarqué comprend des moyens de traitement des réponses aiguillées vers lui par lesdits moyens d'aiguillage de réponses.  
De cette façon, le logiciel client embarqué peut :
  - soit n'effectuer aucune action, les réponses générées par le logiciel principal étant transmises uniquement au logiciel client externe ;
  - soit effectuer un filtrage des réponses qui lui sont transmises sans être transmises au logiciel client externe. Le logiciel client embarqué peut ainsi par exemple décider, en fonction de critères déterminés, quelles réponses doivent être transmises au logiciel client externe (en étant ou non modifiées) et quelles réponses ne doivent pas l'être ;
  - soit un "espionnage" des réponses dont il reçoit une copie et qui sont par ailleurs directement transmises au logiciel client externe.

Préférentiellement, ledit logiciel principal comprend une application principale, basée notamment sur un jeu de fonctions d'exécution, permettant chacune l'exécution d'au moins une desdites commandes de pilotage. Le logiciel client embarqué comprend une application cliente, basée notamment sur un jeu de fonctions sources, permettant chacune l'envoi ou la réception, vers ou en provenance de l'application principale, de commandes de pilotage ou de réponses à des commandes de pilotage. Le logiciel principal et/ou ledit logiciel client embarqué comprend(nent) une application d'interface, permettant l'interfaçage desdites fonctions sources avec lesdites fonctions d'exécution.

Ainsi, on limite les coûts de développement, du fait que l'application d'interface (aussi appelée par la suite "bibliothèque d'interface application"), une fois développée, peut être utilisée avec différentes applications clientes, voire avec différents logiciels principaux.

Dans un mode de réalisation préférentiel de l'invention, ledit jeu de commandes de pilotage est un jeu de commandes AT standard.

5 Ceci permet un développement rapide du logiciel client embarqué, les commandes AT étant bien connues et déjà utilisées pour le développement des logiciels clients externes (hébergés par les équipements tiers). Ceci facilite également le développement d'un logiciel client fortement basé sur un logiciel client externe existant.

10 L'invention concerne également un procédé de mise en œuvre d'un logiciel client de pilotage d'un module de radiocommunication, ledit module de radiocommunication étant du type hébergeant et exécutant un logiciel principal assurant notamment des fonctions de radiocommunication, ledit logiciel principal comprenant des moyens d'exécution de commandes de pilotage, envoyées au logiciel principal par ledit logiciel client de pilotage et appartenant à un jeu de commandes de pilotage prédéterminé. Selon l'invention, ledit module de radiocommunication héberge et exécute au moins un logiciel client, dit logiciel client embarqué. Le logiciel client embarqué et le logiciel principal dialoguent l'un avec l'autre de façon que le logiciel client embarqué joue au moins un  
15 des deux rôles suivants :

- le rôle dudit logiciel client de pilotage, envoyant des commandes de pilotage au logiciel principal, et recevant du logiciel principal des réponses, résultant de l'exécution de certaines des commandes de pilotage ;
- 20 - le rôle d'un logiciel client de supervision, gérant l'exécution de commandes de pilotage envoyées par ledit logiciel client de pilotage, ledit logiciel client de pilotage, dit logiciel client externe, étant hébergé et exécuté par un équipement tiers coopérant avec le module de radiocommunication.

25 D'autres caractéristiques et avantages de l'invention apparaîtront à la lecture de la description suivante d'un mode de réalisation préférentiel de l'invention, donné à titre d'exemple indicatif et non limitatif, et des dessins annexés, dans lesquels :

- la figure 1 présente un schéma simplifié d'un mode de réalisation particulier d'un module de radiocommunication selon la présente invention ;
- la figure 2 illustre le fonctionnement de la technique actuelle de pilotage d'un  
30 module de radiocommunication par un équipement tiers ;



- la figure 3 illustre le fonctionnement d'un premier mode de réalisation du module de radiocommunication selon la présente invention, dans lequel le logiciel client embarqué joue le rôle d'un logiciel client de pilotage ;
- les figures 4 à 7 illustrent chacune un mécanisme de fonctionnement d'un second mode de réalisation du module de radiocommunication selon la présente invention, dans lequel le logiciel client embarqué joue le rôle d'un logiciel client de supervision, et plus précisément :
  - \* la figure 4 illustre un mécanisme d'aiguillage de commandes, permettant au logiciel client embarqué de filtrer les commandes provenant du logiciel client externe ;
  - \* la figure 5 illustre une variante du mécanisme d'aiguillage de commandes de la figure 4, permettant au logiciel client embarqué d'espionner les commandes provenant du logiciel client externe ;
  - \* la figure 6 illustre le mécanisme d'aiguillage de réponses, permettant au logiciel client embarqué de filtrer les réponses destinées au logiciel client externe ;
  - \* la figure 7 illustre une variante du mécanisme d'aiguillage de réponses de la figure 6, permettant au logiciel client embarqué d'espionner les réponses destinées au logiciel client externe.

L'invention concerne donc un module de radiocommunication hébergeant et exécutant, avec le même jeu de ressources (processeur et mémoire), un logiciel principal et (au moins) un logiciel client embarqué.

De façon classique, le logiciel principal (par exemple un "logiciel GSM") assure notamment des fonctions de radiocommunication, et comprend des moyens d'exécution de commandes de pilotage (qui lui envoyées par un logiciel client de pilotage).

Le logiciel client embarqué (concept propre à la présente invention) peut jouer :

- le rôle d'un logiciel client de pilotage (premier mode de réalisation de l'invention, décrit ci-dessous en relation avec la figure 3), et/ou
- le rôle d'un logiciel client de supervision (second mode de réalisation de l'invention, décrit ci-dessous en relation avec les figures 4 à 7).

Dans la suite de la description, on suppose que les commandes de pilotage sont des commandes AT. Il est clair toutefois que la présente invention n'est nullement limitée à ce type de commande de pilotage.

5 Dans le mode de réalisation particulier illustré sur la figure 1, le logiciel principal 3 comprend une application principale 3a, basée notamment sur un jeu de fonctions d'exécution, permettant chacune l'exécution d'au moins une des commandes AT.

Par ailleurs, le logiciel client embarqué 6 comprend :

- 10 - une application cliente 6a, basée notamment sur un jeu de fonctions sources, permettant chacune l'envoi ou la réception, vers ou en provenance de l'application principale 1a, de commandes AT ou de réponses à des commandes AT ;
- une application d'interface 6b, permettant l'interfaçage des fonctions sources (de l'application cliente 6a) avec les fonctions d'exécution (de l'application principale 1a).

15 Pour communiquer l'un avec l'autre, le logiciel client embarqué 6 et le logiciel principal 3 s'appuient chacun sur une couche API, référencées 61 et 31 respectivement. On rappelle que le terme API (pour "Application Programming Interface" en anglais) désigne une interface. C'est une description des règles de communication correspondant à un certain ensemble fonctionnel.

20 Côté logiciel client embarqué 6, la couche API (embarquée) 61 comprend:

- dans l'application cliente embarquée 6a :
  - \* un bloc "Application Mandatory API", formant interface décrivant des fonctions devant être définies dans l'application cliente embarquée ;
- dans l'application d'interface (ou librairie d'interface application) 6b : les trois  
25 blocs suivants :
  - \* un bloc "AT command API", formant interface décrivant l'accès aux commandes AT, cette interface décrit des fonctions se trouvant dans la librairie d'interface application ;
  - \* un bloc "OS API", formant interface décrivant l'accès aux fonctions du  
30 système d'exploitation (Operating System), cette interface décrit des fonctions se trouvant dans la librairie d'interface application ;

- \* un bloc "Standard API", formant interface décrivant l'accès à des fonctions standard, cette interface décrit des fonctions se trouvant dans la librairie d'interface application.

5 Côté logiciel principal 3, la couche API (exportée) 31 comprend, dans l'application principale 3a :

- \* un bloc " AT Command API ", pendant du bloc du même nom de la couche API (embarquée) 61 côté logiciel client embarqué 6 ;
- \* un bloc "OS API, pendant du bloc du même nom de la couche API (embarquée) 61 côté logiciel client embarqué 6.

10 L'application principale 3a constitue le noyau du logiciel principal 3, et l'application cliente 6a et l'application d'interface 6b constituent le noyau 60 du logiciel client embarqué 6.

15 L'application d'interface 6b est une entité logicielle qui permet le dialogue entre les deux entités logicielles autonomes (ou binaires) que sont l'application cliente 6a et l'application principale 1a. Elle se présente par exemple sous la forme d'une librairie (déjà compilée).

20 Le logiciel client embarqué et le logiciel principal utilisent chacun une partie distincte d'une même mémoire vive (RAM). Le client définit la taille de la pile mémoire nécessaire à la bonne exécution du logiciel client embarqué. Une tentative d'accès de l'un des deux logiciels à une partie de la mémoire vive réservée à l'autre logiciel provoque un arrêt de fonctionnement.

On présente maintenant, en relation avec la figure 3, un premier mode de réalisation de l'invention dans lequel le logiciel client embarqué joue le rôle d'un logiciel client de pilotage.

25 Les éléments compris de façon classique dans un module de radiocommunication, et présentés ci-dessus en relation avec la figure 2 (art antérieur), conservent les mêmes références numériques. Ainsi, le module de radiocommunication 1 héberge et exécute un logiciel principal 3 comprenant notamment des moyens 4 d'exécution des commandes AT.

30 Par ailleurs, selon le premier mode de réalisation de l'invention et comme présenté ci-dessus en relation avec la figure 2, le module de radiocommunication 1

héberge et exécute en outre un logiciel client embarqué 6, jouant le rôle d'un logiciel client de pilotage et comprenant notamment une application cliente embarquée 6a et une application d'interface 6b.

Le fonctionnement de ce premier mode de réalisation de l'invention peut être résumé ainsi :

- 5 - étape "1" : l'application cliente embarquée 6a appelle une fonction source (wm\_atSendCommand") d'envoi vers les moyens 4 d'exécution (compris dans le logiciel principal) d'une ou plusieurs commande(s) AT. Cette fonction source est présentée en détail dans la suite de la description ;
- 10 - étape "2" : l'application d'interface 6b appelle la ou les fonctions d'exécution appropriées au sein des moyens 4 d'exécution (compris dans le logiciel principal 3) ;
- étape "3" : les moyens 4 d'exécution exécutent la ou les commande(s) AT ;
- étape "4" : après exécution, les moyens d'exécution 4 envoient la ou les réponse(s) AT à l'application cliente embarquée 6a (si la commande d'envoi précitée  
15 a été paramétrée en ce sens) ;
- étape "5" : cette ou ces réponse(s) est(sont) expédiée(s) par l'application d'interface 6b, qui appelle, au sein de l'application cliente embarquée 6a, la fonction source ("wm\_apmAppliParser") de traitement d'un message provenant du  
20 logiciel principal. Un paramètre de cette fonction source de traitement est le message qui contient la ou les réponse(s) précitée(s). Cette fonction source est présentée en détail dans la suite de la description ;
- étape "6" : au sein de l'application cliente embarquée 6a, la fonction source de traitement traite la réponse.

25 On présente maintenant, en relation avec les figures 4 à 7, un second mode de réalisation de l'invention dans lequel le logiciel client embarqué joue le rôle d'un logiciel client de supervision.

Dans ce second mode de réalisation, le module de radiocommunication n'est pas autonome (contrairement au premier mode de réalisation), mais est piloté par un  
30 équipement tiers avec lequel il coopère. La configuration est donc celle de l'art antérieur, décrite ci-dessus en relation avec la figure 2 (les mêmes éléments conservent les mêmes

références numériques). En d'autres termes, une application cliente externe (logiciel client de pilotage) 2 envoie des commandes AT à un logiciel principal 3, en vue de leur exécution par le logiciel principal 3.

Selon le second mode de réalisation de l'invention, le module de radiocommunication 1 héberge et exécute en outre un logiciel client embarqué 6, jouant le rôle d'un logiciel client de supervision et comprenant notamment une application cliente embarquée 6a et une application d'interface 6b. De façon transparente pour l'application cliente externe 2, le logiciel client embarqué 6 supervise l'exécution (ou non) des commandes AT par les moyens 4 d'exécution compris dans le logiciel principal 3.

Le logiciel client embarqué 6 (logiciel de supervision) peut décider de la mise en œuvre, au sein du module de radiocommunication, notamment de :

- un mécanisme d'aiguillage et de traitement des commandes AT, envoyées par l'application cliente externe (logiciel client de pilotage) 2. Trois variantes de mise en œuvre de ce mécanisme sont par exemple proposées, selon lesquelles le logiciel principal 3 transmet les commandes AT qu'il reçoit : soit uniquement aux moyens 4 d'exécution (première variante), soit uniquement au logiciel client embarqué 6 (seconde variante), soit aux deux (troisième variante) ;
- un mécanisme d'aiguillage et de traitement des réponses AT, résultant de l'exécution par les moyens 4 d'exécution (compris dans le logiciel principal 3) de commandes AT. Trois variantes de mise en œuvre de ce mécanisme sont par exemple proposées, selon lesquelles les réponses AT générées par le logiciel principal 3 sont transmises respectivement uniquement à l'application cliente externe 2 (première variante), uniquement au logiciel client embarqué 6 (seconde variante), ou bien aux deux (troisième variante).

On notera que la première variante de chacun des deux mécanismes précités (relatifs aux commandes AT et aux réponses AT respectivement) signifie que le logiciel client embarqué 6 peut décider d'être totalement passif à certains moments. Elles correspondent donc à un fonctionnement classique du module de radiocommunication, tel que décrit précédemment en relation avec la figure 2, et ne sont pas décrites à nouveau.

On présente maintenant, en relation avec la figure 4, la seconde variante du mécanisme d'aiguillage et de traitement des commandes AT, qui permet au logiciel client embarqué 6 de filtrer les commandes AT provenant de l'application cliente externe 2.

5 Le fonctionnement de cette seconde variante du mécanisme d'aiguillage et de traitement des commandes AT peut être résumé en deux phases successives, à savoir :

- une phase préalable de sélection, par l'application cliente externe 2, de la (seconde) politique d'aiguillage de commandes AT, selon laquelle les commandes AT sont retransmises uniquement au logiciel client embarqué 6, et
- 10 - une phase de traitement, selon la (seconde) politique d'aiguillage de commandes sélectionnée, des commandes AT envoyées par l'application cliente externe 2.

La phase préalable de sélection de la seconde politique d'aiguillage de commandes AT comprend les étapes suivantes :

- étape "1" : l'application cliente embarquée 6a appelle une fonction source ("wm\_atCmdPreParserSubscribe") d'inscription auprès du logiciel principal 3 à un service d'aiguillage de commandes AT (ou "command pre-parsing service" en anglais), avec un paramètre de cette fonction d'inscription qui indique le choix de la seconde politique d'aiguillage de commandes AT. Cette fonction source est présentée en détail dans la suite de la description ;
- 15 - étape "2" : l'application d'interface 6b appelle, au sein du logiciel principal 3, la ou les fonction(s) d'exécution appropriée(s), dite(s) fonction d'enregistrement de l'inscription au service d'aiguillage de commandes AT ;
- 20 - étape "3" : le logiciel principal 3 établit l'inscription demandée par l'application cliente embarquée 6a, via l'application d'interface 6b.
- 25

Dans un seul souci de simplification, on suppose dans la suite de la description (et comme illustré sur la figure 4), que la ou les fonction(s) d'enregistrement de l'inscription au service d'aiguillage de commandes AT sont comprises, au sein du logiciel principal 3, dans les moyens 4 d'exécution de commandes AT.

30

Egalement dans un seul souci de simplification, on suppose dans la suite de la description (et comme illustré sur la figure 4), que les moyens d'aiguillage de commandes (discutés ci-après) sont compris, au sein du logiciel principal 3, dans les moyens 4 d'exécution de commandes AT.

- 5           La phase de traitement des commandes AT comprend les étapes suivantes :
- étape "4" : l'application cliente externe 2 envoie une commande AT au logiciel principal 3 ;
  - étape "5" : la liaison série 5 transmet la commande AT à des moyens d'aiguillage de commandes, compris dans les moyens 4 d'exécution (eux-mêmes compris dans le logiciel principal 3) et fonctionnant selon la seconde politique d'aiguillage de commandes AT (sélectionnée lors de la phase préalable) ;
  - 10   - étape "6" : sans être exécutées par les moyens 4 d'exécution, la commande AT est retransmise uniquement au logiciel client embarqué 6 ;
  - étape "7" : la commande AT est expédiée par l'application d'interface 6b, qui
  - 15   appelle, au sein de l'application cliente embarquée 6a, la fonction source ("wm\_apmAppliParser") de traitement d'un message provenant du logiciel principal, paramétrée ici notamment par un message qui contient la commande AT et qui indique qu'il s'agit de la commande AT "originale". Cette fonction source est présentée en détail dans la suite de
  - 20   la description ;
  - étape "8" : au sein de l'application cliente embarquée 6a, la fonction source de traitement traite la commande AT.

25           Ce traitement consiste par exemple à renvoyer la commande AT vers les moyens 4 d'exécution (selon le mécanisme correspondant au premier mode de réalisation de l'invention, décrit ci-dessus en relation avec la figure 3). Il peut également consister en la fourniture arbitraire d'une réponse par le logiciel client embarqué 6 lui-même, sans que la commande AT soit exécutée. Dans ce cas, le logiciel client embarqué 6 tient par exemple compte d'au moins une information relative à la commande AT concernée (type de commande, nature du ou des paramètres, etc). D'une façon générale, quel que

30   soit le traitement effectué, on comprend que le logiciel client embarqué 6 "filtre" la commande AT.

On présente maintenant, en relation avec la figure 5, la troisième variante du mécanisme d'aiguillage et de traitement des commandes AT, qui permet au logiciel client embarqué 6 d'espionner les commandes AT provenant de l'application cliente externe 2.

5           Le fonctionnement de cette troisième variante du mécanisme d'aiguillage et de traitement des commandes AT peut aussi être résumé en deux phases successives, à savoir:

- une phase préalable de sélection, par l'application cliente externe 2, de la (troisième) politique d'aiguillage de commandes AT, selon laquelle les  
10       commandes AT sont retransmises non seulement au logiciel client embarqué 6 mais aussi aux moyens 4 d'exécution (compris dans le logiciel principal 3), et
- une phase de traitement, selon la (troisième) politique d'aiguillage de commandes sélectionnée, des commandes AT envoyées par l'application cliente externe 2.

15       Le fonctionnement de cette troisième variante diffère de celui de la seconde variante essentiellement en ce que :

- lors de l'étape "1" de la phase préalable, l'application cliente embarquée 6a sélectionne la troisième (et non pas la seconde) politique d'aiguillage de commandes AT ;
- lors de l'étape "6" de la phase de traitement, la commande AT est transmise aux  
20       moyens 4 d'exécution et une copie de cette commande AT est transmise au logiciel client embarqué 6 ;
- lors de l'étape "8" de la phase de traitement, au sein de l'application cliente embarquée 6a, la fonction source de traitement traite la copie de la commande AT ;
- la phase de traitement comprend en outre une étape "7", au cours de laquelle les  
25       moyens 4 d'exécution compris dans le logiciel principal 3 exécutent la commande AT.

30       On présente maintenant, en relation avec la figure 6, la seconde variante du mécanisme d'aiguillage et de traitement des réponses AT, qui permet au logiciel client embarqué 6 de filtrer les réponses AT destinées à l'application cliente externe 2.



Le fonctionnement de cette seconde variante du mécanisme d'aiguillage et de traitement des réponses AT peut aussi être résumé en deux phases successives, à savoir :

- une phase préalable de sélection, par l'application cliente externe 2, de la (seconde) politique d'aiguillage de réponses AT, selon laquelle les réponses AT générées par le logiciel principal 3 sont transmises uniquement au logiciel client embarqué 6 ;
- une phase de traitement, selon la (seconde) politique d'aiguillage de réponses sélectionnée, des réponses AT générées par le logiciel principal 3.

La phase préalable de sélection de la seconde politique d'aiguillage de réponses AT comprend les étapes suivantes :

- étape "1" : l'application cliente embarquée 6a appelle une fonction source ("wm\_atRspPreParserSubscribe") d'inscription auprès du logiciel principal 3 à un service d'aiguillage de réponses AT (ou "response pre-parsing service" en anglais), avec un paramètre de cette fonction d'inscription qui indique le choix de la seconde politique d'aiguillage de réponses AT. Cette fonction source est présentée en détail dans la suite de la description ;
- étape "2" : l'application d'interface 6b appelle, au sein du logiciel principal 3, la ou les fonction(s) d'exécution appropriée(s), dite(s) fonction d'enregistrement de l'inscription au service d'aiguillage de réponses AT ;
- étape "3" : le logiciel principal 3 établit l'inscription demandée par l'application cliente embarquée 6a, via l'application d'interface 6b.

Dans un seul souci de simplification, on suppose dans la suite de la description (et comme illustré sur la figure 6), que la ou les fonction(s) d'enregistrement de l'inscription au service d'aiguillage de réponse AT sont comprises, au sein du logiciel principal 3, dans les moyens 4 d'exécution de commandes AT.

Egalement dans un seul souci de simplification, on suppose dans la suite de la description (et comme illustré sur la figure 6), que les moyens d'aiguillage de réponses (discutés ci-après) sont compris, au sein du logiciel principal 3, dans les moyens 4 d'exécution de commandes AT.

La phase de traitement des réponses AT comprend les étapes suivantes :

- étape "4" : l'application cliente externe 2 envoie une commande AT au logiciel principal 3 ;
- étape "5" : la liaison série 5 transmet la commande AT aux moyens 4 d'exécution (compris dans le logiciel principal 3) ;
- 5 - étape "6" : les moyens 4 d'exécution exécutent la commande AT et génère une réponse AT ;
- étape "7" : des moyens d'aiguillage de réponses, compris dans les moyens 4 d'exécution et fonctionnant selon la seconde politique d'aiguillage de réponses AT (sélectionnée lors de la phase préalable), envoient la réponse AT au logiciel client embarqué 6 ;
- 10 - étape "8" : la réponse AT est expédiée par l'application d'interface 6b, qui appelle, au sein de l'application cliente embarquée 6a, la fonction source ("wm\_apmAppliParser") de traitement d'un message provenant du logiciel principal, paramétrée ici notamment par un message qui contient la réponse AT et qui indique qu'il s'agit de la réponse AT "originale" ;
- 15 - étape "9" : au sein de l'application cliente embarquée 6a, la fonction source de traitement traite la réponse AT. On peut parler ici de "filtrage" des réponses AT par le logiciel client embarqué.

On présente maintenant, en relation avec la figure 7, la troisième variante du mécanisme d'aiguillage et de traitement des réponses AT, qui permet au logiciel client embarqué 6 d'espionner les réponses AT destinées à l'application cliente externe 2.

Le fonctionnement de cette troisième variante du mécanisme d'aiguillage et de traitement des réponses AT peut aussi être résumé en deux phases successives, à savoir:

- une phase préalable de sélection, par l'application cliente externe 2, de la (troisième) politique d'aiguillage de réponses AT, selon laquelle les réponses AT sont retransmises non seulement au logiciel client embarqué 6 mais aussi à l'application cliente externe 2, et
- 25 - une phase de traitement, selon la (troisième) politique d'aiguillage de réponses sélectionnée, des réponses AT générées par le logiciel principal 3.

30 Le fonctionnement de cette troisième variante diffère de celui de la seconde variante essentiellement en ce que :

- lors de l'étape "1" de la phase préalable, l'application cliente embarquée 6a sélectionne la troisième (et non pas la seconde) politique d'aiguillage de réponses AT ;
- 5       - lors de l'étape "7" de la phase de traitement, la réponse AT est transmise à l'application cliente externe 2 et une copie de cette réponse AT est transmise vers le logiciel client embarqué 6 ;
- lors de l'étape "9" de la phase de traitement, au sein de l'application cliente embarquée 6a, la fonction source de traitement traite la copie de la réponses AT ;
- 10       - la phase de traitement comprend en outre une étape "8", au cours de laquelle la réponse est envoyée à travers la liaison série 5, et une étape "9" au cours de laquelle l'application cliente externe 2 reçoit et traite la réponse.

On trouvera en Annexe 1 une présentation détaillée de certaines des fonctions sources sur lesquelles est basée l'application cliente embarquée.

- 15       A titre illustratif, on trouvera en Annexe 2 un exemple d'application embarquée, écrite en langage C et dont l'objectif est d'envoyer une commande AT13 trois secondes après chaque initialisation et écriture d'objets en mémoire Flash.

Dans le cadre de la présente invention, de façon optionnelle, on peut également prévoir d'utiliser, outre les commandes AT standard, au moins une des commandes AT supplémentaires suivantes :

- 20       - commande de chargement d'un logiciel client (AT+WDWL), permettant au logiciel client externe de charger un nouveau logiciel client dans le module de radiocommunication ;
- commande de désactivation (AT+WOPEN), permettant au logiciel client externe de désactiver le logiciel client embarqué.

## Présentation détaillée de certaines fonctions sources sur lesquelles est basée l'application cliente embarquée

fonction de traitement d'un message provenant du logiciel principal. Le message formant paramètre de la fonction de traitement contient notamment une commande AT ou une réponse à une commande AT.

```
bool wm_apmAppliParser(wm_apmMsg_t * Message);
```

15      **Message**

La structure du message change avec chaque type de message reçu :

```
typedef struct
{
    20      s16                      MsgTyp;
    /* "MsgTyp" est un type de message reçu, qui permet de déterminer la structure
    associée du corps du message */
    wm_apmBody_t    Body;      /* "Body" est un corps de message spécifique */
} wm_apmMsg_t;
```

25

Valeurs de "MsgTyp" :

WM AT SEND RSP

30 Le message contient une réponse à une commande AT préalablement envoyée  
au logiciel principal par le logiciel client embarqué

WM\_AT\_UNSOLICITED

Le message contient une commande AT non sollicitée

35

WM\_AT\_CMD\_PRE\_PARSER

Le message contient une commande AT envoyée par un logiciel client externe,  
via le logiciel principal

40

WM AT RSP PRE PARSER

Le message contient une réponse AT résultant de l'exécution par le logiciel principal d'une commande AT provenant d'une application externe.

#### WM\_OS\_TIMER

5 Le message est envoyé à l'expiration d'une temporisation.

La structure du corps est :

```

10 typedef union
{
    /* Sont incluses ici toutes les structures spécifiques associées aux types de
    message "MsgTyp" */
    /* WM_AT_SEND_RSP */
    15 wm_atResponse_t          ATResponse;

    /* WM_AT_UNSOLICITED */
    wm_atUnsolicited_t        ATUnsolicited;

    /* WM_AT_CMD_PRE_PARSER */
    20 wm_atCmdPreParser_t      ATCmdPreParser;

    /* WM_AT_RSP_PRE_PARSER */
    wm_atRspPreParser_t        ATRspPreParser

    25 /* WM_OS_TIMER */
    wm_osTimer_t              OSTimer;
} wm_apmBody_t;
```

Les sous-structures du corps sont :

```

30 Corps pour WM_AT_SEND_RSP :
    typedef struct {
        wm_atSendRspType_e    Type;
        u16                   StrLength;    /* Longueur de strData */
        35 char                 StrData[1];  /* réponse AT */
    } wm_atResponse_t;

    typedef enum {
        WM_AT_SEND_RSP_TO_EMBEDDED,
        40 WM_AT_SEND_RSP_TO_EXTERNAL,
        WM_AT_SEND_RSP_BROADCAST
    } wm_atSendRspType_e;

    (voir le détail de la fonction "wm_atSendCommand", pour la description de
    45 "wm_atSendRspType_e description").
```

**Corps pour WM\_AT\_UNSOLICITED :**

```
typedef struct {  
    wm_atUnsolicited_e    Type;  
    u16                    StrLength;  
    char                    StrData[1];  
} wm_atUnsolicited_t;
```

```
typedef enum {  
    WM_AT_UNSOLICITED_TO_EXTERNAL,  
    WM_AT_UNSOLICITED_TO_EMBEDDED,  
    WM_AT_UNSOLICITED_BROADCAST  
} wm_atUnsolicited_e;
```

(voir le détail de la fonction "wm\_atUnsolicitedSubscription", pour la description de "wm\_atUnsolicited\_e").

**Corps pour WM\_AT\_CMD\_PRE\_PARSER :**

```
typedef struct {  
    wm_atCmdPreSubscribe_e Type;  
    u16                    StrLength;  
    char                    StrData[1];  
} wm_atCmdPreParser_t;
```

```
typedef enum {  
    WM_AT_CMD_PRE_WAVECOM_TREATMENT, /* Valeur par défaut */  
    WM_AT_CMD_PRE_EMBEDDED_TREATMENT,  
    WM_AT_CMD_PRE_BROADCAST  
} wm_atCmdPreSubscribe_e;
```

(voir le détail de la fonction "wm\_atRspPreParserSubscribe", pour la description de "wm\_atCmdPreSubscribe\_e").

35

**Corps pour WM\_AT\_RSP\_PRE\_PARSER :**

```

typedef struct {
    wm_atRspPreSubscribe_e Type;
    u16                      StrLength;
    char                    StrData[1];
5 } wm_atRspPreParser_t;

typedef enum {
    WM_AT_RSP_PRE_WAVECOM_TREATMENT, /* Valeur par défaut */
    WM_AT_RSP_PRE_EMBEDDED_TREATMENT,
10 WM_AT_RSP_PRE_BROADCAST
} wm_atRspPreSubscribe_e;

(voir le détail de la fonction "wm_atRspPreParserSubscribe", pour la description
de "wm_atRspPreSubscribe_e").
15

```

#### Corps pour WM\_OS\_TIMER :

```

typedef struct {
    u8          Ident;          /* Identifiant de Temporisation */
20 } wm_osTimer_t;

(voir le détail de la fonction "wm_osStartTimer", pour la description de "Ident").

```

#### Valeurs retournées

Le paramètre de retour indique si le message a été traité (VRAI) ou non (FAUX).

#### A2) "wm\_atSendCommand"

fonction d'envoi vers le logiciel principal d'au moins une commande AT, dont un paramètre indique le(s) logiciel(s) destinataire(s) (à savoir le logiciel client embarqué et/ou le logiciel client externe) de la réponse résultant de l'exécution de cette commande AT.

```

Nom exact :
void wm_atSendCommand ( u16          wm_atSendRspType_e  AtStringSize,
35                               char    ResponseType,
                                   *AtString, );

```

#### Paramètres

AtString

Ce paramètre peut être tout type de chaîne de commandes AT, en caractères ASCII.  
Plusieurs chaînes peuvent être envoyées en même temps.

#### AtStringSize

5 Taille du paramètre précédent : AtString.

#### ResponseType

Type de la réponse

```

10 typedef enum {
    WM_AT_SEND_RSP_TO_EMBEDDED, /* Valeur par défaut */
    WM_AT_SEND_RSP_TO_EXTERNAL,
    WM_AT_SEND_RSP_BROADCAST
} wm_atSendRspType_e;

15 WM_AT_SEND_RSP_TO_EMBEDDED
    Toutes les réponses sont redirigées vers l'application cliente embarquée
    (Embedded application). C'est le mode par défaut.

20 WM_AT_SEND_RSP_TO_EXTERNAL
    Toutes les réponses sont redirigées vers l'application cliente externe (PC).

    WM_AT_SEND_RSP_BROADCAST
    Toutes les réponses sont redirigées ("diffusées") vers l'application cliente
    embarquée et l'application cliente externe (PC).

```

25

#### A3) "wm\_atUnsolicitedSubscription" :

fonction d'inscription auprès du logiciel principal à un service de réception de  
commandes AT non sollicitées, dont un paramètre indique vers quel(s) logiciel(s)  
destinataire(s) (à savoir le logiciel client embarqué et/ou le logiciel client externe) doit  
30 être redirigée chacune des commandes AT non sollicitées

#### Nom exact :

```

void wm_atUnsolicitedSubscription (
35     wm_atUnsolicited_e Unsolicited);

```

#### Paramètres

##### Unsolicited

40 Ce paramètre décrit l'action effectuée quand une commande AT non sollicitée arrive.

```

typedef enum {
    WM_AT_UNSOLICITED_TO_EXTERNAL, /* Valeur par défaut */
    WM_AT_UNSOLICITED_TO_EMBEDDED,
    WM_AT_UNSOLICITED_BROADCAST,
45 } wm_atUnsolicited_e;

```

##### WM\_AT\_UNSOLICITED\_TO\_EXTERNAL

Toutes les commandes non sollicitées seront redirigées vers l'application cliente  
externe (PC) (mode par défaut)



**WM\_AT\_UNSOLICITED\_TO\_EMBEDDED**

Toutes les commandes non sollicitées seront redirigées vers l'application cliente embarquée

5

**WM\_AT\_UNSOLICITED\_BROADCAST**

Toutes les commandes non sollicitées seront redirigées (diffusées) vers l'application cliente externe (PC) et l'application cliente embarquée

10 **A4) "wm\_atCmdPreParserSubscribe" :**

fonction d'inscription auprès du logiciel principal à un service d'aiguillage de commandes AT, dont un paramètre indique vers quel(s) logiciel(s) (à savoir le logiciel principal et/ou le logiciel client embarqué) doit être aiguillée chacune des commandes AT provenant d'une application externe.

15

Nom exact :

```
void      wm_atCmdPreParserSubscribe (
                                wm_atCmdPreSubscribe_e SubscribeType);
```

20

**Paramètres****SubscribeType**

Ce paramètre décrit l'action effectuée quand une commande AT arrive

25

```
typedef enum      {
    WM_AT_CMD_PRE_WAVECOM_TREATMENT, /* Valeur par défaut */
    WM_AT_CMD_PRE_EMBEDDED_TREATMENT,
    WM_AT_CMD_PRE_BROADCAST
} wm_atCmdPreSubscribe_e;
```

30

**WM\_AT\_CMD\_PRE\_WAVECOM\_TREATMENT**

L'application cliente embarquée ne veut pas filtrer (ni espionner) les commandes envoyées par l'application cliente externe (mode par défaut).

35

**WM\_AT\_CMD\_PRE\_EMBEDDED\_TREATMENT**

L'application cliente embarquée veut filtrer les commandes envoyées par l'application cliente externe

**WM\_AT\_CMD\_PRE\_BROADCAST**

L'application cliente embarquée veut espionner les commandes envoyées par l'application cliente externe

40

**A5) "wm\_atRspPreParserSubscribe" :**

fonction d'inscription auprès du logiciel principal à un service d'aiguillage de réponses, dont un paramètre indique vers quel(s) logiciel(s) (à savoir le logiciel client externe et/ou le logiciel client embarqué) doit être aiguillée chacune des réponses résultant de

45

l'exécution par le logiciel principal d'une commande AT provenant d'une application externe.

Nom exact :

```
5 void wm_atRspPreParserSubscribe (
    wm_atRspPreSubscribe_e SubscribeType);
```

Paramètres

10 SubscribeType

Ce paramètre décrit l'action effectuée quand une commande AT arrive

```
15 typedef enum {
    WM_AT_RSP_PRE_WAVECOM_TREATMENT, /* Valeur par défaut */
    WM_AT_RSP_PRE_EMBEDDED_TREATMENT,
    WM_AT_RSP_PRE_BROADCAST
} wm_atRspPreSubscribe_e;
```

20 WM\_AT\_RSP\_PRE\_WAVECOM\_TREATMENT

L'application cliente embarquée ne veut pas filtrer (ni espionner) les réponses envoyées à l'application cliente externe (mode par défaut).

25 WM\_AT\_RSP\_PRE\_EMBEDDED\_TREATMENT

L'application cliente embarquée veut filtrer les réponses envoyées à l'application cliente externe.

WM\_AT\_RSP\_PRE\_BROADCAST

L'application cliente embarquée veut espionner les réponses envoyées à l'application cliente externe.

30

**A6) "wm\_atSendRspExternalApp" :**

fonction d'envoi vers le logiciel client externe, via le logiciel principal, d'au moins une réponse. L'utilisation de cette fonction n'est possible que si une inscription préalable au service d'aiguillage de réponses a été effectuée, avec aiguillage d'une copie des réponses

35 notamment vers le logiciel client embarqué

Nom exact :

```
40 void wm_atSendRspExternalApp (u16 char AtStringSize,
    *AtString);
```

Paramètres

AtString

45 peut être tout type de chaîne de réponses AT, en caractères ASCII

AtStringSize

Taille du paramètre précédent : AtString

**A7) "DataFlowService" :**

5 fonction d'émission et/ou réception de données par le logiciel client embarqué, via le logiciel principal, après qu'une communication de données a été établie

**A8) "wm\_osStartTimer" :**

10 fonction d'armement d'une temporisation, un paramètre de ladite fonction d'armement indiquant la temporisation à armer

**A9) "wm\_osStopTimer" :**

fonction d'arrêt d'une temporisation, un paramètre de ladite fonction d'arrêt indiquant la temporisation à arrêter

**ANNEXE 2 :****Exemple d'application cliente embarquée**

```

5  /* File      : Appli.c */
   /*-----*/
   /* Object    : Customer application */
   /*           : This program waits 3 seconds, send a AT13 command to */
10  /*           : software version. It starts a cyclic timer of 3 seconds */
   /*           : and writes an object in Fash with a value of 0. */
   /*           : Every 3 seconds, it reads object value, display it, */
   /*           : increments and rewrites it in Flash. */
   /*-----*/
15  /* contents  : Customer main procedures */
   /*-----*/
   /* Change    : */
   /*-----*/
   /* Date      | Author    | Modification */
   /*-----+-----+-----*/
20  /* 26/12/00 | FCA       | Creation */
   /*-----+-----+-----*/
   /*-----*/

#include "wm_types.h"
#include "wm_apm.h"
25 #include "utils.h"

#define TIMER_INIT      01
#define TIMER_CYCLIC 02
30 #define OBJECT_ID      0x1

char wm_apmCustomStack[1024];
const ul6 wm_apmCustomStackSize = sizeof(wm_apmCustomStack);

35  /*-----*/
   /* Function   : wm_apmAppliInit */
   /*-----*/
   /* Objet     : Customer application initialisation */
   /*-----*/
40  /* Return    : */
   /*-----*/
   /*-----*/
   /* Variable Name | IN | OUT | GLB | Utilisation */
   /*-----+-----+-----+-----+-----*/
45  /* InitType     |   |   |   | Application start mode reason */
   /*-----+-----+-----+-----+-----*/
   /*-----*/
void wm_apmAppliInit (wm_apmInitType_e InitType)
{
50     // Trace : Level 10
     wm_osDebugTrace(10, "Init : Start timer" );

     // Start a timer for 3 seconds
     wm_osStartTimer (TIMER_INIT,FALSE, 30);
55 }

   /*-----*/
   /* Function   : wm_apmAppliParser */
   /*-----*/
   /*-----*/
60  /* Objet     : Customer task body */
   /*-----*/

```

```

/* Return      ;                                     */
/*                                                     */
/*-----*/
/* Variable Name      |IN |OUT|GLB|  Utilisation                                     */
/*-----*/
/* pMessage           |  |  |  |  Message sent by real time kernel                                     */
/*-----*/
/*****
bool wm_apmAppliParser (wm_apmMsg_t * pMessage)
{
    char strCommand [] = "ATI3\r"; // don't forget '\r' at the end
    u16  u16DataLen;
    bool bRet;
    char strTraceMsg[50];
    char strData;

    switch (pMessage->MsgTyp)
    {
        case WM_OS_TIMER :
            switch (pMessage->Body.OSTimer.Ident)
            {
                case TIMER_INIT: // Timer armed in wm_apmAppliInit
                    wm_osDebugTrace(10, "TIMER_INIT" );

                    // Result will be visible in the AT terminal window
                    wm_atSendCommand(wm_strlen(strCommand)+1,
                                    WM_AT_SEND_RSP_TO_EXTERNAL, strCommand);

                    // Write object for the first time
                    strData = 0x30;

                    bRet = wm_osWriteFlashData(OBJECT_ID, sizeof(char),
                                                &strData );

                    if (bRet)
                        wm_osStartTimer (TIMER_CYCLIC,TRUE, 30);
                    else
                        wm_osDebugTrace(10, "wm_osWriteFlashData failed" );
                    break;

                case TIMER_CYCLIC:
                    wm_osDebugTrace(10, "TIMER_CYCLIC" );
                    // Read object, increment value, rewrite object
                    u16DataLen = wm_osGetLenFlashData(OBJECT_ID);
                    u16DataLen = wm_osReadFlashData(OBJECT_ID, u16DataLen,
                                                    &strData );
                    if (u16DataLen)
                    {
                        // Print object after read
                        wm_strcpy(strTraceMsg, "Value :");
                        strTraceMsg[w_m_strlen(strTraceMsg)] = strData;
                        strTraceMsg[w_m_strlen(strTraceMsg)+1] = '\0';

                        wm_osDebugTrace(10, strTraceMsg );

                        // increment it
                        strData++;
                        if (strData == 0x3A) strData = 0x30;

                        bRet = wm_osWriteFlashData(OBJECT_ID, sizeof(char),
                                                    &strData );

                        if (!bRet)

```

5

10

15

20

```
        wm_osDebugTrace(10, "wm_osWriteFlashData failed" );  
    }  
    else  
        wm_osDebugTrace(10, "Read 0 byte" );  
  
        break;  
  
        default:  
            break;  
    }  
  
    default:  
        break;  
}  
return TRUE;  
}
```

### REVENDEICATIONS

1. Module de radiocommunication, du type hébergeant et exécutant un logiciel principal assurant notamment des fonctions de radiocommunication, ledit logiciel principal comprenant des moyens d'exécution de commandes de pilotage, envoyées au logiciel principal par au moins un logiciel client de pilotage et appartenant à un jeu de commandes de pilotage prédéterminé,
- 5 caractérisé en ce que ledit module de radiocommunication héberge et exécute en outre au moins un logiciel client, dit logiciel client embarqué,
- et en ce que le logiciel client embarqué et le logiciel principal comprennent des moyens permettant au logiciel client embarqué de jouer au moins un des deux rôles suivants :
- 10
- le rôle d'un logiciel client de pilotage, envoyant des commandes de pilotage au logiciel principal, et recevant du logiciel principal des réponses, résultant de l'exécution de certaines des commandes de pilotage ;
  - le rôle d'un logiciel client de supervision, gérant l'exécution de commandes de pilotage envoyées par un logiciel client de pilotage, dit logiciel client externe,

15 hébergé et exécuté par un équipement tiers coopérant avec le module de radiocommunication.
2. Module de radiocommunication selon la revendication 1, caractérisé en ce que, pour permettre au logiciel client embarqué de jouer le rôle d'un logiciel client de pilotage :
- 20
- le logiciel client embarqué comprend des moyens d'envoi de commandes de pilotage vers les moyens d'exécution compris dans le logiciel principal ;
  - le logiciel principal comprend des moyens d'envoi des réponses, résultant de l'exécution de certaines des commandes de pilotage par les moyens d'exécution

25 compris dans le logiciel principal, vers le logiciel client embarqué ;

  - le logiciel client embarqué comprend des moyens de traitement des réponses envoyées vers lui par le logiciel principal.
3. Module de radiocommunication selon l'une quelconque des revendications 1 et 2, caractérisé en ce que, pour permettre au logiciel client embarqué de jouer le rôle d'un
- 30 logiciel client de supervision :

- le logiciel principal comprend des moyens d'aiguillage de commandes, en fonction d'une politique d'aiguillage de commandes déterminée, de façon à transmettre les commandes de pilotage provenant du logiciel client externe vers le logiciel client embarqué et/ou vers les moyens d'exécution compris dans le logiciel principal ;
  - le logiciel client embarqué comprend des moyens de traitement des commandes de pilotage aiguillées vers lui par lesdits moyens d'aiguillage de commandes.
4. Module de radiocommunication selon la revendication 3, caractérisé en ce que le logiciel client embarqué comprend des moyens de sélection de la politique d'aiguillage de commandes appliquée par lesdits moyens d'aiguillage de commandes, parmi un ensemble de politiques d'aiguillage de commandes telles que respectivement :
- les commandes de pilotage provenant du logiciel client externe sont transmises uniquement aux moyens d'exécution compris dans le logiciel principal ;
  - les commandes de pilotage provenant du logiciel client externe sont transmises uniquement au logiciel client embarqué ;
  - les commandes de pilotage provenant du logiciel client externe sont transmises aux moyens d'exécution compris dans le logiciel principal et au logiciel client embarqué.
5. Module de radiocommunication selon l'une quelconque des revendications 3 et 4, caractérisé en ce que lesdits moyens de traitement des commandes prennent, pour chaque commande, au moins une décision appartenant au groupe comprenant :
- envoi de la commande de pilotage vers les moyens d'exécution compris dans le logiciel principal, le logiciel client embarqué comprenant à cet effet des moyens d'envoi de commandes de pilotage vers les moyens d'exécution ;
  - fourniture ou non d'une réponse, uniquement en fonction d'au moins une information relative à la commande, sans exécution de la commande, le logiciel client embarqué comprenant à cet effet des moyens d'envoi de la réponse au logiciel client externe, via le logiciel principal.
6. Module de radiocommunication selon l'une quelconque des revendications 1 à 5, caractérisé en ce que, pour permettre au logiciel client embarqué de jouer le rôle d'un logiciel client de supervision :



- le logiciel principal comprend des moyens d'aiguillage de réponses, en fonction d'une politique d'aiguillage de réponses déterminée, de façon à transmettre des réponses, résultant de l'exécution de certaines des commandes de pilotage par les moyens d'exécution compris dans le logiciel principal, vers le logiciel client embarqué et/ou vers le logiciel client externe;
  - le logiciel client embarqué comprend des moyens de traitement des réponses aiguillées vers lui par lesdits moyens d'aiguillage de réponses.
7. Module de radiocommunication selon la revendication 6, caractérisé en ce que le logiciel client embarqué comprend des moyens de sélection de la politique d'aiguillage de réponses appliquée par lesdits moyens d'aiguillage de réponses, parmi un ensemble de politiques d'aiguillage de réponses telles que respectivement :
- les réponses provenant des moyens d'exécution sont transmises uniquement au logiciel client externe ;
  - les réponses provenant des moyens d'exécution sont transmises uniquement au logiciel client embarqué ;
  - les réponses provenant des moyens d'exécution sont transmises au logiciel client embarqué et au logiciel client externe.
8. Module de radiocommunication selon l'une quelconque des revendications 7 et 8, caractérisé en ce qu'il est compris dans un dispositif appartenant au groupe comprenant :
- des terminaux de radiocommunication ;
  - des dispositifs, autres que des terminaux de radiocommunication, nécessitant une fonctionnalité de communication sans fil ;
  - des modems.
9. Module de radiocommunication selon l'une quelconque des revendications 1 à 8, caractérisé en ce que ledit logiciel principal comprend une application principale, basée notamment sur un jeu de fonctions d'exécution, permettant chacune l'exécution d'au moins une desdites commandes de pilotage,
- en ce que ledit logiciel client embarqué comprend une application cliente, basée notamment sur un jeu de fonctions sources, permettant chacune l'envoi ou la réception, vers ou en provenance de l'application principale, de commandes de pilotage ou de réponses à des commandes de pilotage,

et en ce que ledit logiciel principal et/ou ledit logiciel client embarqué comprend(nent) une application d'interface, permettant l'interfaçage desdites fonctions sources avec lesdites fonctions d'exécution.

5       **10.**     Module de radiocommunication selon l'une quelconque des revendications 1 à 9, caractérisé en ce que ledit logiciel client embarqué comprend une application cliente, basée notamment sur un jeu de fonctions sources, permettant chacune l'envoi ou la réception, vers ou en provenance de l'application principale, de commandes de pilotage ou de réponses à des commandes de pilotage, et en ce que ledit jeu de fonctions sources comprend notamment une fonction  
10     ("wm\_apmAppliParser") de traitement d'un message provenant du logiciel principal, un paramètre de ladite fonction de traitement étant ledit message.

**11.**     Module de radiocommunication selon la revendication 10, caractérisé en ce que ledit message formant paramètre de ladite fonction de traitement possède une structure comprenant :

- 15       -       un premier champ contenant une information relative au type dudit message ;  
-       un second champ contenant le corps spécifique dudit message.

**12.**     Module de radiocommunication selon la revendication 11, caractérisé en ce que le type dudit message appartient au groupe comprenant :

- 20       -       message contenant une réponse à une commande de pilotage préalablement envoyée au logiciel principal par le logiciel client embarqué ;  
-       message contenant une commande de pilotage non sollicitée ;  
-       message contenant une commande de pilotage envoyée par un logiciel client externe, via le logiciel principal ;  
-       message contenant une réponse résultant de l'exécution par le logiciel principal  
25       d'une commande de pilotage ;  
-       message envoyé à l'expiration d'une temporisation.

**13.**     Module de radiocommunication selon l'une quelconque des revendications 10 à 12, caractérisé en ce que ledit jeu de fonctions sources comprend en outre au moins une fonction appartenant au groupe comprenant :

- 30       -       une fonction ("wm\_atSendCommand") d'envoi vers le logiciel principal d'au moins une commande de pilotage, un premier paramètre de ladite fonction

d'envoi étant ladite au moins une commande de pilotage, un second paramètre de ladite fonction d'envoi indiquant le(s) logiciel(s) destinataire(s), à savoir le logiciel client embarqué et/ou le logiciel client externe, de la réponse résultant de l'exécution de ladite commande de pilotage ;

- 5        -        une fonction ("wm\_atUnsolicitedSubscription") d'inscription auprès du logiciel principal à un service de réception de commandes de pilotage non sollicitées, un paramètre de ladite fonction d'inscription indiquant vers quel(s) logiciel(s) destinataire(s), à savoir le logiciel client embarqué et/ou le logiciel client externe, doit être redirigée chacune des commandes de pilotage non sollicitées ;
- 10       -        une fonction ("wm\_atCmdPreParserSubscribe") d'inscription auprès du logiciel principal à un service d'aiguillage de commandes de pilotage, un paramètre de ladite fonction d'inscription indiquant vers quel(s) logiciel(s), à savoir le logiciel principal et/ou le logiciel client embarqué, doit être aiguillée chacune des commandes de pilotage provenant du logiciel client externe ;
- 15       -        une fonction ("wm\_atRspPreParserSubscribe") d'inscription auprès du logiciel principal à un service d'aiguillage de réponses, un paramètre de ladite fonction d'inscription indiquant vers quel(s) logiciel(s), à savoir le logiciel client externe et/ou le logiciel client embarqué, doit être aiguillée chacune des réponses résultant de l'exécution par le logiciel principal d'une commande de pilotage ;
- 20       -        une fonction ("wm\_atSendRspExternalApp") d'envoi vers le logiciel client externe, via le logiciel principal, d'au moins une réponse, un paramètre de ladite fonction d'envoi étant ladite au moins une réponse.

14.    Module de radiocommunication selon l'une quelconque des revendications 1 à 13, caractérisé en ce que le logiciel client embarqué et le logiciel principal utilisent  
25       chacun une partie distincte d'une même mémoire vive, une tentative d'accès de l'un des logiciels à une partie de la mémoire vive réservée à l'autre logiciel provoquant un arrêt de fonctionnement.

15.    Module de radiocommunication selon l'une quelconque des revendications 1 à 14, caractérisé en ce que ledit jeu de commandes de pilotage est un jeu de commandes  
30       AT standard.

16. Module de radiocommunication selon la revendication 15, caractérisé en ce que ledit jeu de commandes de pilotage comprend, outre les commandes AT standard, une commande AT supplémentaire, dite commande de chargement d'un logiciel client (AT+WDWL), permettant au logiciel client externe de charger un nouveau logiciel client dans le module de radiocommunication.

17. Module de radiocommunication selon l'une quelconque des revendications 15 et 16, caractérisé en ce que ledit jeu de commandes de pilotage comprend, outre les commandes AT standard, une commande AT supplémentaire, dite commande de désactivation (AT+WOPEN), permettant au logiciel client externe de désactiver le logiciel client embarqué.

18. Procédé de mise en œuvre d'un logiciel client de pilotage d'un module de radiocommunication, ledit module de radiocommunication étant du type hébergeant et exécutant un logiciel principal assurant notamment des fonctions de radiocommunication, ledit logiciel principal comprenant des moyens d'exécution de commandes de pilotage, envoyées au logiciel principal par ledit logiciel client de pilotage et appartenant à un jeu de commandes de pilotage prédéterminé, caractérisé en ce que ledit module de radiocommunication héberge et exécute au moins un logiciel client, dit logiciel client embarqué,

et en ce que le logiciel client embarqué et le logiciel principal dialoguent l'un avec l'autre de façon que le logiciel client embarqué joue au moins un des deux rôles suivants :

- le rôle dudit logiciel client de pilotage, envoyant des commandes de pilotage au logiciel principal, et recevant du logiciel principal des réponses, résultant de l'exécution de certaines des commandes de pilotage ;
- le rôle d'un logiciel client de supervision, gérant l'exécution de commandes de pilotage envoyées par ledit logiciel client de pilotage, ledit logiciel client de pilotage, dit logiciel client externe, étant hébergé et exécuté par un équipement tiers coopérant avec le module de radiocommunication.

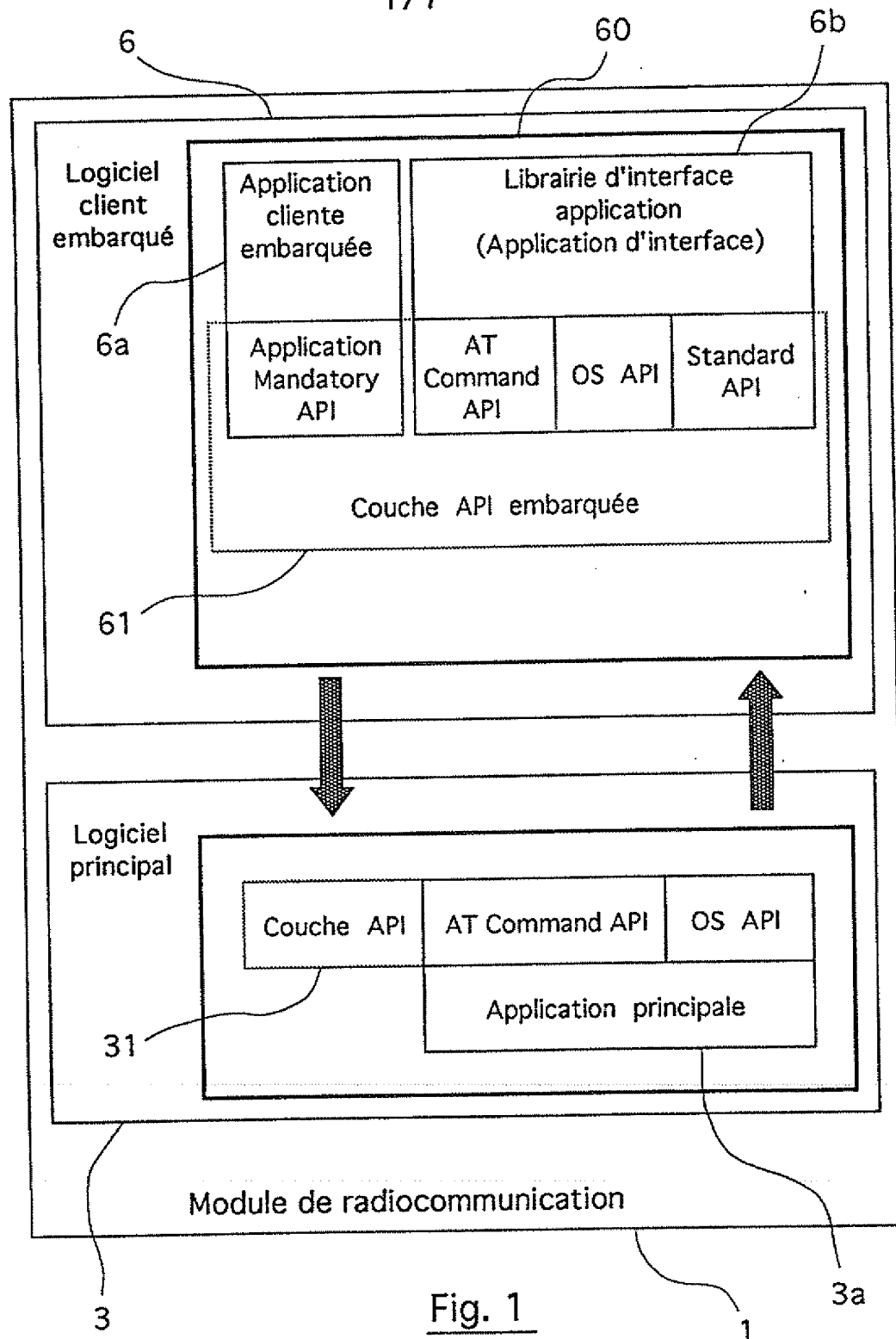
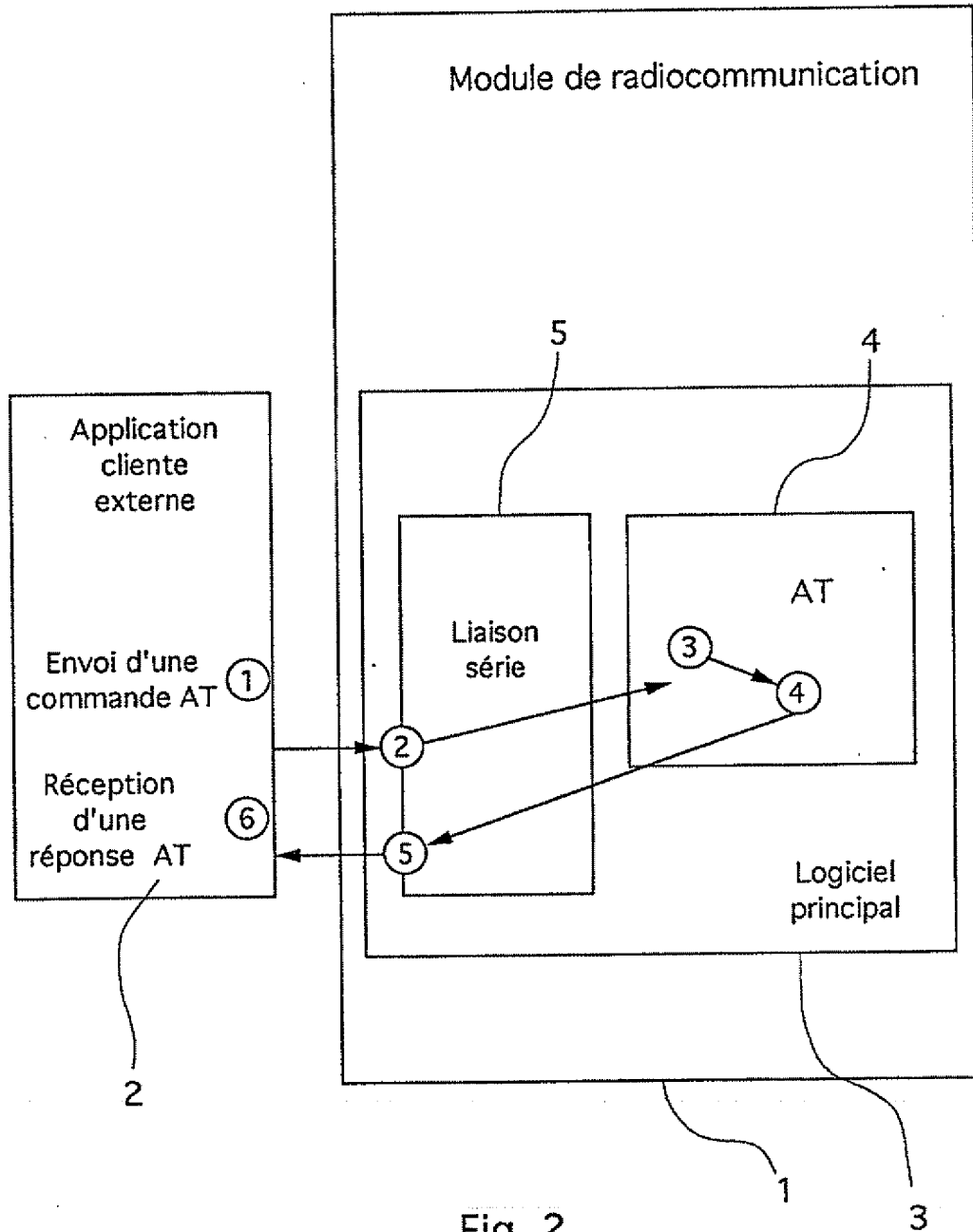
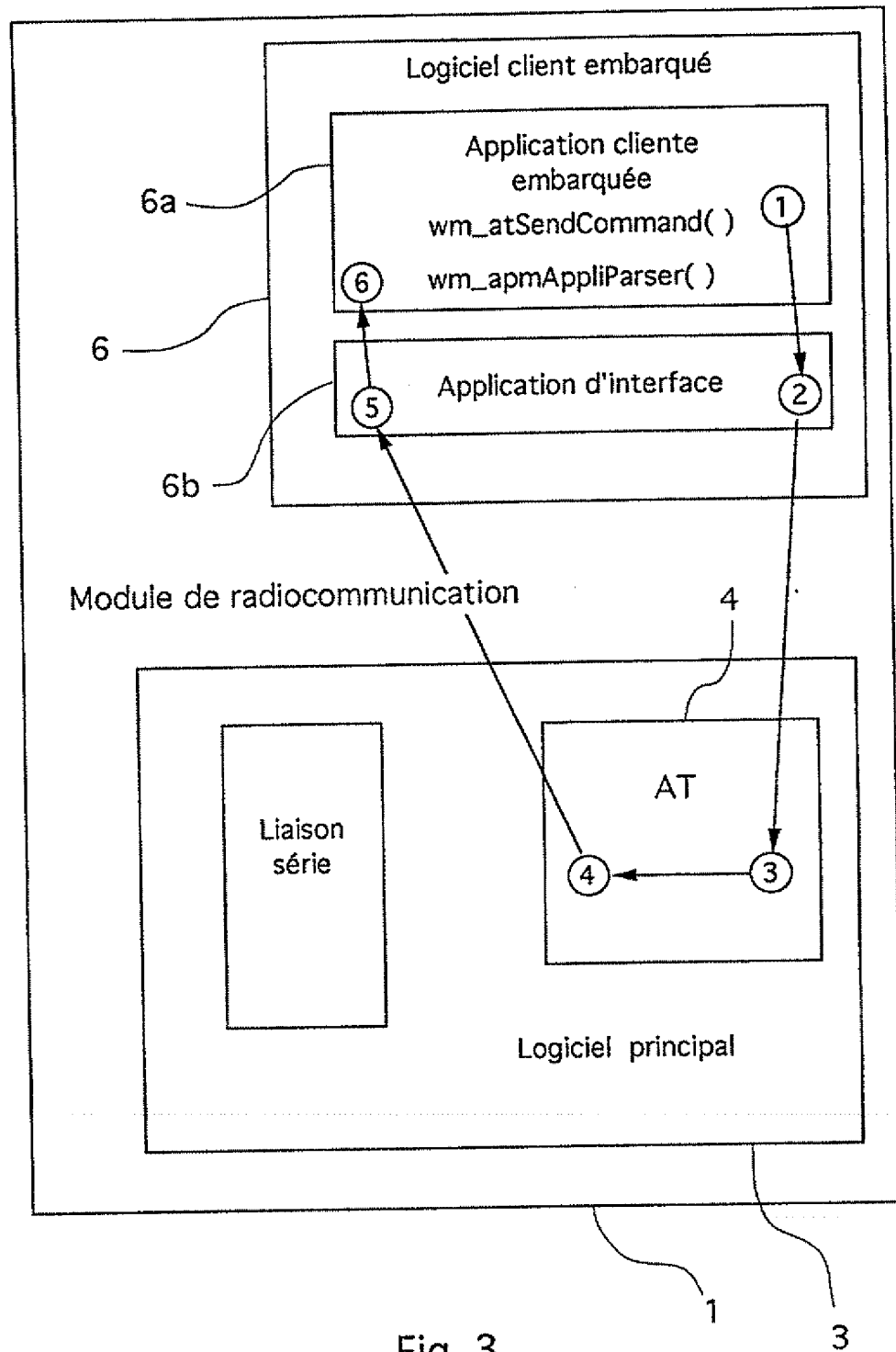


Fig. 1

Fig. 2

( ART ANTERIEUR )

Fig. 3

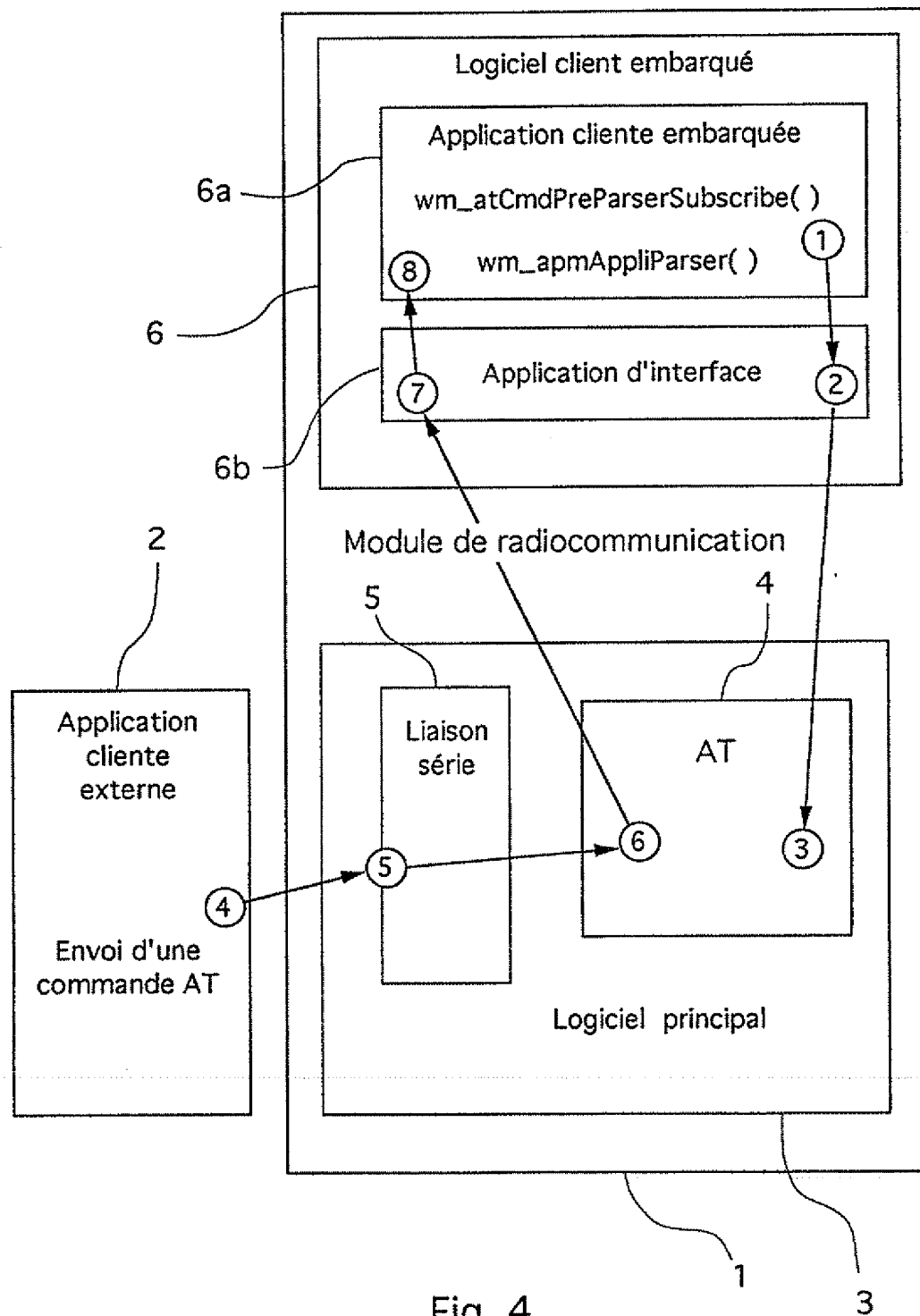


Fig. 4



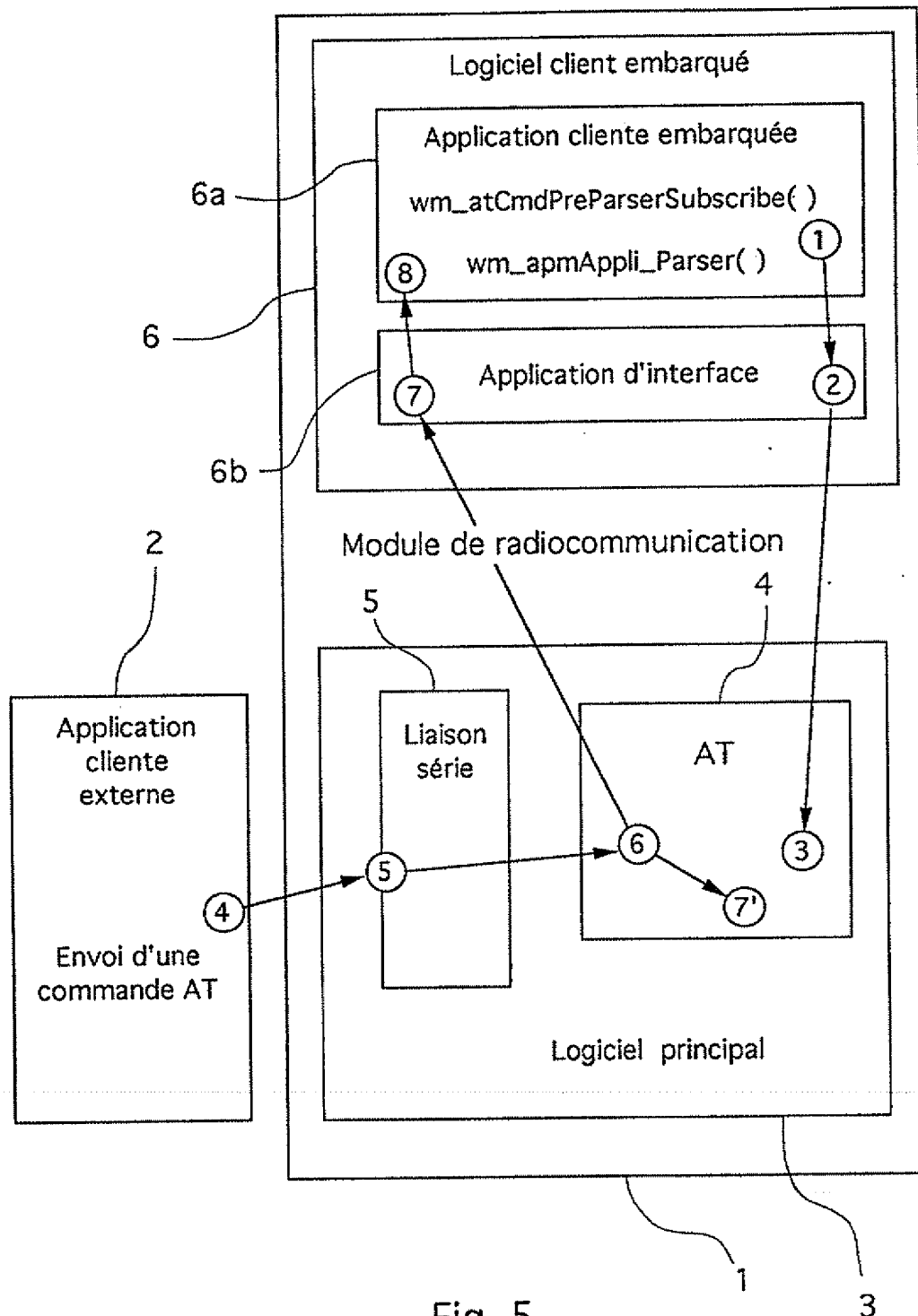


Fig. 5

6/7

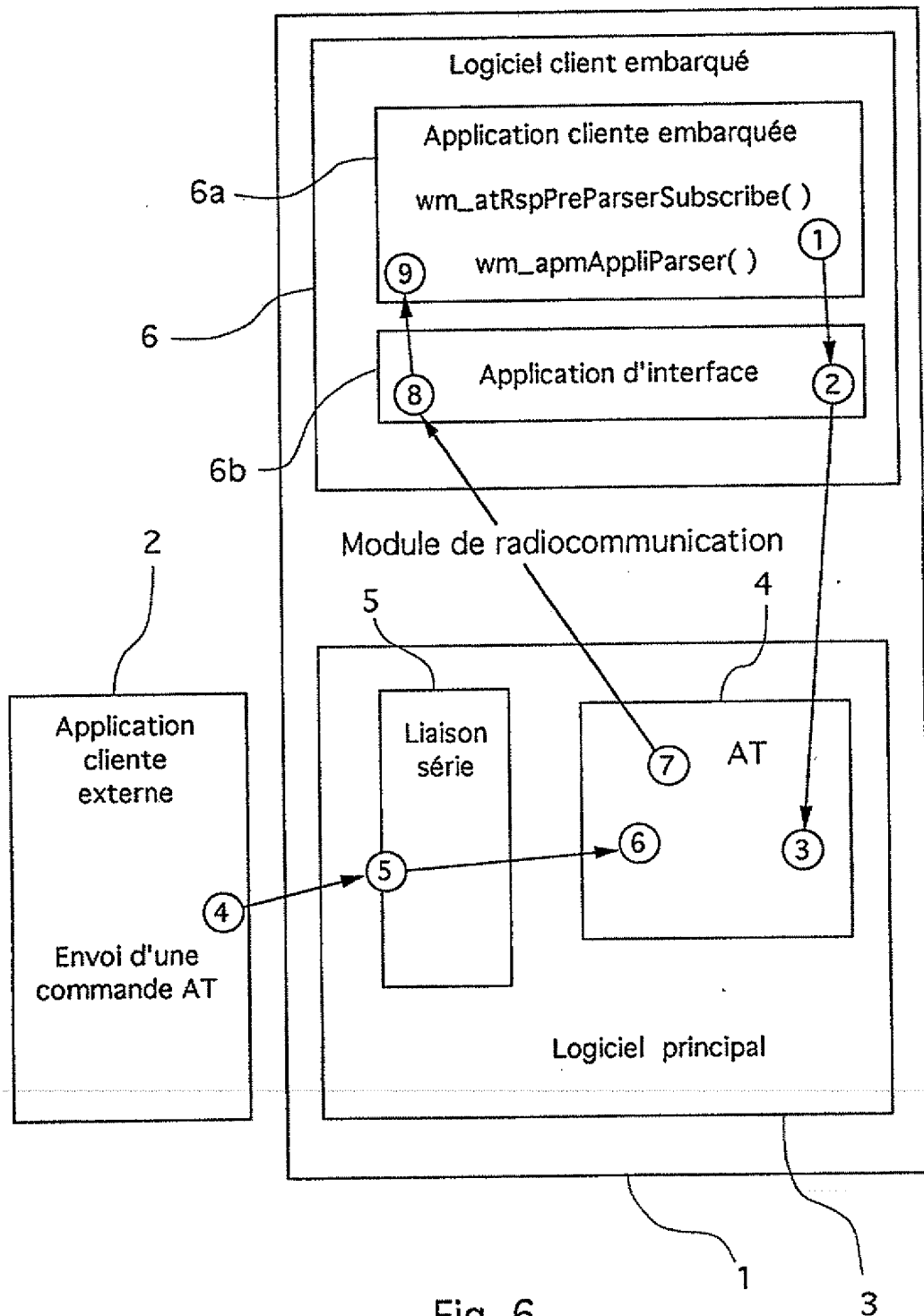


Fig. 6